

Amendments to the Specification:

Please insert the following new paragraphs after paragraph [0019]:

[0019.1] Fig. 8 is a flowchart of an embodiment of a process for dynamic port management.

[0019.2] Fig. 9 is a flowchart of an embodiment of a process for reconciling two communication sessions requesting the same registered network address.

Please replace paragraph [0032] with the following amended paragraph:

[0032] This “setup” process may use a plurality of packets communicated between the DPM driver 42c and the DPM server 44c. For instance, any given packet 52 will have a header section 52a. In these packets, the source IP address/port will still be IPx/123 as assigned by the DPM driver, however the destination IP address is now an unregistered IP address of the DPM server IPy, and the port is fixed to a predetermined one of the gateway such as a “well-known” port 1080. The information about the true/final destination (e.g., the destination computer 20) is embedded in a data section 52b of the packet which should include at least, in this case, IP_{out}:23 and an indicator about the replaceability of the port number, as indicated by data field 53. It is understood that since this destination information and port replaceability is contained in the data section of the packet, not the header section, various methods can be implemented to have both the DPM driver and server to agree on a predetermined mechanism for each of them to extract such information.

Please replace paragraph [0034] with the following amended paragraph:

[0034] Referring now to Fig. 7 and also Fig. 8, a flow diagram 70 and a more detailed flowchart 90 summarize ~~summarizes~~ the steps taken by the DPM driver and DPM server for manipulating the IP address and port for an application session according to one embodiment of the present invention. Before all the steps are taken, it is assumed that each computer or server is loaded with DPM driver software and the gateway is equipped with DPM server software.

Execution begins at step 72, where an application session (communication) is initiated from the DPM driver, and an initial source port is obtained (step 91 in Fig. 8). At step 74, a setup process is executed between the DPM driver and DPM server to inform the DPM server about the final destination IP address and its corresponding port. The DPM server then obtains a port number for the session (step 92 in Fig. 8). At step 76, the DPM server checks to see whether the port number may be changed (step 93 in Fig. 8). This may be done by examining data field 53 in Fig. 6. If so, execution proceeds to step 78 where the DPM server selects an available port ~~and a registered IP address~~, and dynamically assigns this port for all outgoing packets in the session (step 94 in Fig. 8). Once the port is established (either at steps 76 or 78), execution proceeds to step 80 where the DPM server finds an available registered IP address for outgoing packets (step 95 in Fig. 8). To complete the header of the data packet, the destination network address and port number are extracted and included in the data packet (steps 96 and 97 in Fig. 8). The reserved registered network address and the first or second port number (if the port number is not replaced or is replaced) are included in the data packet as the source address and port number (step 98 in Fig. 8). At step 82, a look-up table, which may be stored in the gateway 18, is then updated to reflect the one-to-one relationship between the pair of the originating source IP address (e.g., an unregistered IP address) and the initial port of the application session and the pair of the outgoing registered IP address assigned by the gateway 18 and the dynamically assigned port (step 99 in Fig. 8).

Please replace paragraph [0036] with the following amended paragraph:

[0036] Referring to Fig. 9 for a flowchart of an embodiment of a process for reconciling two competing sessions. For example, consider that a first FTP session request is from address IP_x and initial port 123 (step 100), and a second one is from address IP_z and port 123 (step 101). Further, both sessions request the use of a registered address of IP₂. The first session has a destination FTP server at address IP₁ and port 23, and the second session has a destination FTP server at address IP₃ and port 23. In this case, the DPM server will still allow the same registered IP address IP₂ and port 123 to be used concurrently by the two FTP sessions (step 102) and the

look-up table is updated accordingly (step 103). The reason that the DPM server can do so is because the look-up table can distinguish the two sessions by two different destination IP addresses (i.e., IP₁ and IP₃) for future communications. For example, the look-up table for this example may show:

	<u>SOURCE ADDRESS</u>	<u>SOURCE PORT NO.</u>	<u>ASSIGNED SOURCE ADDR.</u>	<u>ASSIGNED SOURCE PORT NO.</u>	<u>DEST. ADDRESS</u>	<u>DEST. PORT NO.</u>
<u>REQUEST 1</u>	<u>IP_x</u>	<u>123</u>	<u>IP₂</u>	<u>123</u>	<u>IP₁</u>	<u>23</u>
<u>REQUEST 2</u>	<u>IP_y</u>	<u>123</u>	<u>IP₂</u>	<u>123</u>	<u>IP₃</u>	<u>23</u>

When a return packet comes back from one of the destination FTP servers (step 106), although it is targeted for IP₂ and port 123, it can be identified and routed appropriately base on the fact that the IP addresses of the FTP servers can be differentiated, and that the look-up table provides the unique unregistered IP addresses of the computer inside the private network for further packet routing (steps 107 and 108). If at least one of the initial ports (e.g., 123) can be modified, a new port can be dynamically assigned to replace the initial port. From the perspective of the look-up table, the one-to-one relation between the DPM driver and server can more easily be identified since there is at least one more “differentiator” (i.e., the port used by DPM server for outgoing packets) available as compared to the situation where neither one of the ports are changeable.